



Non-interactive ADAS and fundamental data

Martin O'Mullane
Department of Physics
University of Strathclyde

Fundamental data production and ADAS

One of the philosophies of ADAS is to provide baseline quality data for atomic processes of any ion stage of an arbitrary element.

These are selectively updated with higher quality data.

Fundamental data production codes available within ADAS:

- ▶ Cowan code to produce *adf04* data.
- ▶ AUTOSTUCTURE for dielectronic recombination. *
- ▶ CADW ionisation rate generation.
- ▶ R-matrix for electron impact excitation data. *
- ▶ IDL and FORTRAN routines for ECIP, Lodge ion impact and others.
- For charge exchange data extraction from a universal formula.

* Difficult to class these as a baseline quality code.

adas701 — AUTOSTRUCTURE (for dielectronic recombination)

Developed by Nigel Badnell and based on work by W Eissner this (originally a structure code) is tuned to produce state selective dielectronic data.

Data File Name: /home/mog/adas_dev/adas/adf27/dr/helike/mom93#he/n5ls12-n.dat

Title for Run:

Directory for AUTOSTRUCTURE file output:
/home/a_giunta/drm_prep/adas204_run/

- Default file choice -

Select type of run : Structure Satellite Rates

Activate the following files:

<input type="checkbox"/> olg	<input type="checkbox"/> olg	<input type="checkbox"/> radwin	<input type="checkbox"/> radwin
<input type="checkbox"/> ols	<input type="checkbox"/> ols	<input type="checkbox"/> radout	<input type="checkbox"/> radout
<input type="checkbox"/> oic	<input type="checkbox"/> pic	<input type="checkbox"/> TERMS	<input type="checkbox"/> TERMS
<input type="checkbox"/> olsu	<input type="checkbox"/> olsu	<input type="checkbox"/> LEVELS	<input type="checkbox"/> LEVELS
<input type="checkbox"/> oicu	<input type="checkbox"/> picu	<input type="checkbox"/> opls	<input type="checkbox"/> opls
<input type="checkbox"/> RESTART	<input type="checkbox"/> RESTART	<input type="checkbox"/> opic	<input type="checkbox"/> opic

Warning: One or more files already exist.

Text Output Replace Default File Name

File Name : /home/mog/adas_dev/wortk/develop_adas807

Data from adas701

- ▶ AUTOSTRUCTURE produces the data for the DR Project.
- ▶ 13 isoelectronic sequences, elements up to Zn, IC and LS resolutions.
- ▶ 1.2Gb data in *adf09* collection.
- ▶ Sets very high bar for ‘baseline’ data.
- ▶ See N R Badnell *et al*, ‘Dielectronic recombination data for dynamic finite-density plasmas’, *Astron & Astrophys.*, 406, 1151–1165 (2003).
- ▶ Extended for photo-excitation and ionisation.

Input to adas701

A little daunting so reading the manual is essential.

From /home/adas/adas/adf27/dr/olike/oiz00#o/cu12ic23-n.dat:

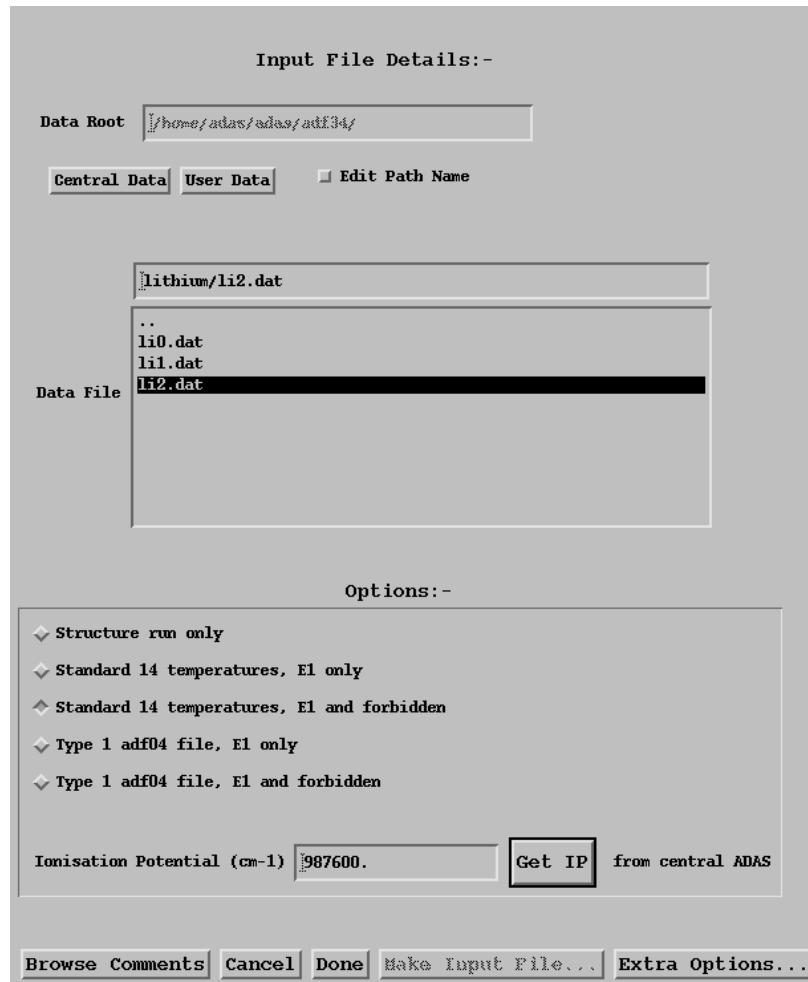
S.S.

```
123456789 22533514517 22533515517 22533516517  
          12543514517 12543515517 12543516517  
          22543518 22543519 2254351A 2254351B 2254351C  
          12553518 12553519 1255351A 1255351B 1255351C  
          22533514518 22533514519 2253351451A 2253351451B 2253351451C  
          22533515518 22533515519 2253351551A 2253351551B 2253351551C  
          22533516518 22533516519 2253351651A 2253351651B 2253351651C  
          12543514518 12543514519 1254351451A 1254351451B 1254351451C  
          12543515518 12543515519 1254351551A 1254351551B 1254351551C  
          12543516518 12543516519 1254351651A 1254351651B 1254351651C  
          22543514 22543515 22543516 22543517  
          12553514 12553515 12553516 12553517  
          22553 12563  
          710720721730731732802900901902903904
```

```
&SALGEB  RUN='DR' RAD='YES' CUP='IC' KORB1=1 KORB2=1 MSTART=4 &END  
&DRR      NMIN=4 NMAX=15 JND=14 LMIN=0 LMAX=6 LCON=5 &END  
          16   20   25   35   45   55   70   100  150   200  300   450  700   999  
&SMINIM  NZION=29 PRINT='UNFORM'  &END  
&SRADWIN KEY=-9  &END  
&SRADCON MENG=-15 &END  
          0.0000 160.0000
```

adas801 — Cowan code for adf04 production

This was the first fundamental data generation code added to the structure and forms the basis of series 8.



adas8#1 — Cowan code for adf04 production

The # indicates an offline code.

- ▶ Designed to be independent of the interactive system (and hence IDL).
- ▶ Self-contained to be portable to large computer systems.
- ▶ Workhorse code for the heavy species project.

```
/home/adas/offline_adas/adas8#1/scripts/run_adas8#1 c2.in c2.inst c2.pp
```

The input file can be crafted by hand or IDL based tools from the heavy species project can be used.

As an example consider Sn^{13+}

- ▶ What is its ground state configuration?
- ▶ What configurations contribute to spectral emission?
- ▶ And to radiated power?
- ▶ How do we choose which ones to include?

The *adf00* set archives ionisation potential and ground configurations:

tin	-50	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	10010	10011	10012	10013	10014	10015	10016	10017	10018	10019	10020	10021	10022	10023	10024	10025	10026	10027	10028	10029	10030	10031	10032	10033	10034	10035	10036	10037	10038	10039	10040	10041	10042	10043	10044	10045	10046	10047	10048	10049	10050	10051	10052	10053	10054	10055	10056	10057	10058	10059	10060	10061	10062	10063	10064	10065	10066	10067	10068	10069	10070	10071	10072	10073	10074	10075	10076	10077	10078	10079	10080	10081	10082	10083	10084	10085	10086	10087	10088	10089	10090	10091	10092	10093	10094	10095	10096	10097	10098	10099	100100	100101	100102	100103	100104	100105	100106	100107	100108	100109	100110	100111	100112	100113	100114	100115	100116	100117	100118	100119	100120	100121	100122	100123	100124	100125	100126	100127	100128	100129	100130	100131	100132	100133	100134	100135	100136	100137	100138	100139	100140	100141	100142	100143	100144	100145	100146	100147	100148	100149	100150	100151	100152	100153	100154	100155	100156	100157	100158	100159	100160	100161	100162	100163	100164	100165	100166	100167	100168	100169	100170	100171	100172	100173	100174	100175	100176	100177	100178	100179	100180	100181	100182	100183	100184	100185	100186	100187	100188	100189	100190	100191	100192	100193	100194	100195	100196	100197	100198	100199	100200	100201	100202	100203	100204	100205	100206	100207	100208	100209	100210	100211	100212	100213	100214	100215	100216	100217	100218	100219	100220	100221	100222	100223	100224	100225	100226	100227	100228	100229	100230	100231	100232	100233	100234	100235	100236	100237	100238	100239	100240	100241	100242	100243	100244	100245	100246	100247	100248	100249	100250	100251	100252	100253	100254	100255	100256	100257	100258	100259	100260	100261	100262	100263	100264	100265

What configurations should be considered?

With a ground state of $3d^{10}4s^24p^64d^1$ we can

- ▶ promote the valence 4d electron to any higher nl shell
- ▶ allow 4s or 4p electrons to be excited
- ▶ or any other electron — from 2p perhaps?
- ▶ however where do we stop in Δn or Δl ?
- ▶ and how many configurations should we consider?

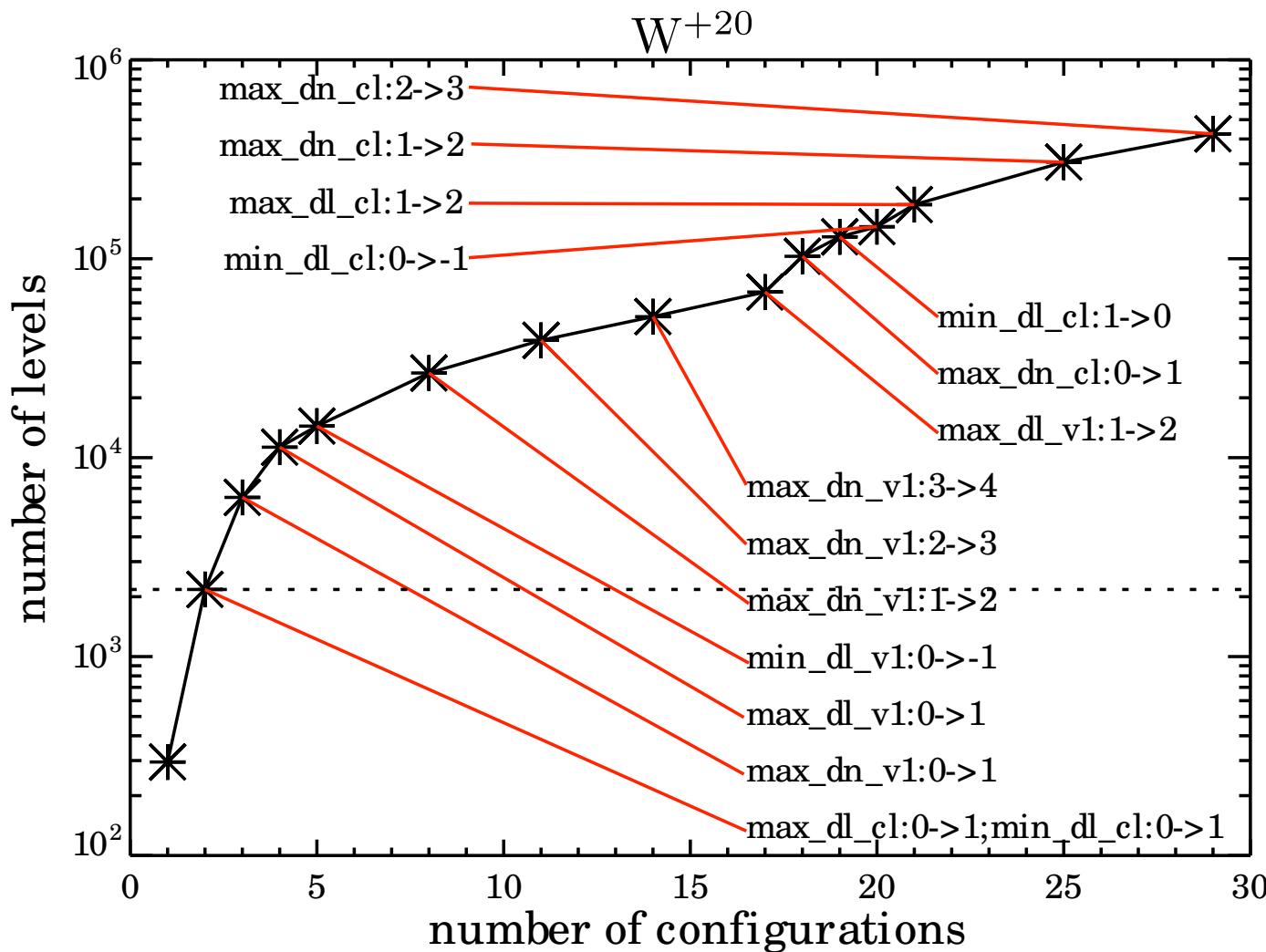
There are 180 distinct ground configurations (for elements up to Radon)

A rule based method is desirable (essential!)

ADAS rules for choosing where to promote electrons

<i>index[]</i>	: index of ground configuration of each ion of element in <i>adf54</i> file
<i>config[]</i>	: ground configuration for each ion of element
<i>n_el[]</i>	: number of electrons for each ion of element
<i>no_v_shl[]</i>	: number of open (valence) shells. Include outer-most shell even if closed.
<i>max_dn_v1[]</i>	: maximum Δn promotion for first (outer-most) valence shell.
<i>min_dn_v1[]</i>	: minimum Δn promotion for first (outer-most) valence shell. Negative value allows access to inner unoccupied or open shells
<i>max_dl_v1[]</i>	: maximum delta Δl promotion for first (outer-most) valence shell.
<i>min_dl_v1[]</i>	: minimum delta Δl promotion for first (outer-most) valence shell.
<i>max_dn_v2[]</i>	: maximum Δn promotion for second (inner-most) valence shell.
<i>min_dn_v2[]</i>	: maximum Δn promotion for second (inner-most) valence shell.
<i>max_dl_v2[]</i>	: maximum delta Δl promotion for second (inner-most) valence shell.
<i>min_dl_v2[]</i>	: minimum delta Δl promotion for second (inner-most) valence shell.
<i>prom_cl[]</i>	: promote from inner shell closed shells (1=yes,0=no).
<i>max_n_cl[]</i>	: maximum inner shell <i>n</i> from which promotions are permitted.
<i>min_n_cl[]</i>	: minimum inner shell <i>n</i> from which promotions are permitted.
<i>max_l_cl[]</i>	: maximum inner shell <i>l</i> from which promotions are permitted.
<i>min_l_cl[]</i>	: minimum inner shell <i>l</i> from which promotions are permitted.
<i>max_dn_cl[]</i>	: maximum Δn promotion from a permitted inner shell.
<i>min_dn_cl[]</i>	: minimum Δn promotion from a permitted inner shell. Negative values of Δn allow access to inner unoccupied or open shells.
<i>max_dl_cl[]</i>	: maximum Δl promotion from a permitted inner shell.
<i>min_dl_cl[]</i>	: minimum Δl promotion from a permitted inner shell.
<i>fill_n_v1[]</i>	: add all <i>nl</i> configurations of outer valence shell <i>n</i> (1=yes,0=no).
<i>fill_par[]</i>	: if <i>n_fill</i> only add opposite parity to valence shell else add both parities (1=yes, 0=n0).
<i>for_tr_sel[]</i>	: Cowan option for radiative transitions 1 - first parity, 2 or 3(default).
<i>last_4f[]</i>	: shift an electron valence shell to unfilled 4f as extra ground.
<i>grd_cmplx[]</i>	: include configurations of same complex as ground configuration for valence <i>n</i> -shell.

adf54 : rules for automatic data generation



Care needed!! resolved calculations (ic or LS) can overwhelm computers.

Work through Sn¹³⁺

- ▶ Within ADAS the generation of heavy species data is almost exclusively a non-GUI activity.
- ▶ The outputs are standard *adf11*, *adf15* and *adf40* datasets which can be used and examined with the GUI interactive system.

At the IDL command line:

```
; Let's choose Sn13+
```

```
z_nuc = 50  
z_ion = 13  
tag = xxesym(z_nuc, /lower) + string(z_ion, format='(i2.2)')
```

```
; Use promotion rules from W work
```

```
a54file = '/home/adas/adas/adf54/promotion_rules_w_adf54.dat'
```

```

adas8xx_promotion_rules, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip,  $
                                prom_rules=rules, a54file = file
help, rules, /st

** Structure <9b54e9c>, 25 tags, length=60, data length=60, refs=1:
CONFIG      STRING ' 1s2  2s2  2p6  3s2  3p6  3d10 4s2  4p6  4d1'
INDEX       INT    129
NO_V_SHL   INT    1
MAX_DN_V1  INT    3
MIN_DN_V1  INT    0
MAX_DL_V1  INT    2
MIN_DL_V1  INT   -2
MAX_DN_V2  INT    0
MIN_DN_V2  INT    0
MAX_DL_V2  INT    0
MIN_DL_V2  INT    0
PROM_CL    INT    1
MAX_N_CL   INT    4
MIN_N_CL   INT    4

```

MAX_L_CL	INT	1
MIN_L_CL	INT	0
MAX_DN_CL	INT	1
MIN_DN_CL	INT	0
MAX_DL_CL	INT	2
MIN_DL_CL	INT	0
FILL_N_V1	INT	1
FILL_PAR	INT	0
FOR_TR_SEL	INT	3
LAST_4F	INT	0
GRD_CMPLX	INT	0

```

adas8xx_promotions, z0_nuc = z_nuc, z_ion = z_ion, ionpot = ip,  $
                           prom_rules          = rules,           $
                           promotion_results = results

help, results, /st

** Structure <9b530dc>, 11 tags, length=2496, data length=2496, refs=1:
   GRD_CFG           STRING    '4d1  '
   GRD_OCC           INT       Array[36]
   EX_CFG            STRING    Array[25]
   GRD_PAR           INT       0
   EX_PAR            INT       Array[25]
   GRD_ZC_COW        LONG      -14
   EX_ZC_COW         LONG      Array[25]
   OC_STORE          INT       Array[36, 26]
   NO_CONFIGS        LONG      Array[7]
   NO_TERMS          LONG      Array[7]
   NO_LEVELS         LONG      Array[7]

```

```
print, results.grd_occ
```

2	2	6	2	6	10	2	6	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

```
print, results.oc_store[:,1]
```

2	2	6	2	6	10	2	6	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

```
print, results.oc_store[:,2]
```

2	2	6	2	6	10	2	6	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

```

; Write CA driver files for restricted plasma parameters

files = { adf34_file      : 'adf34_ic_' + tag + '.dat', $  

          adf42_ic_file : 'adf42_ic_' + tag + '.dat', $  

          adf04_ic_file : 'adf04_ic_' + tag + '.dat', $  

          adf40_ic_file : 'adf40_ic_' + tag + '.dat', $  

          adf15_ic_file : 'adf15_ic_' + tag + '.dat', $  

          adf11_ic_file : 'adf11_ic_' + tag + '.dat' }

plasma = {theta           : [ 1.0e3, 2.0e3, 5.0e3, 1.0e4, 1.5e4, $  

                               2.0e4, 5.0e4, 1.0e5], $  

          idx_theta       : indgen(8),  

          rho            : [ 1.0e8, 1.0e10, 1.0e12, 1.0e14],  

          idx_rho         : indgen(4),  

          npix           : [ 128,    256],  

          wvlmin         : [100.0,   1.0],  

          wvlmax         : [150.0, 500.0],  

          idx_wvl        : indgen(2),  

          theta_noscale : 0,  

          rho_scale      : 0

```

```
adas8xx_create_drivers, z0_nuc=z_nuc, z_ion=z_ion, ionpot=ip, $  
promotion_results=results, $  
plasma=plasma, files=files
```

The driver file for ADAS801 (Cowan code):

2	-5	2	10	1.0	5.d-09	5.d-11-2	0130	1.0	0.65	0.0	0.5
50	-14	Sn	ground	z1=13	0	4d1					
50	-14	Sn	cfg	01	0	5s1					
50	-14	Sn	cfg	02	0	5d1					
50	-14	Sn	cfg	03	0	5g1					
50	-14	Sn	cfg	04	0	6s1					
50	-14	Sn	cfg	05	0	6d1					
50	-14	Sn	cfg	06	0	6g1					
50	-14	Sn	cfg	07	0	7s1					
50	-14	Sn	cfg	08	0	7d1					
50	-14	Sn	cfg	09	0	7g1					
50	-32	Sn	cfg	10	0	3d10	4s1	4p6	4d2		
50	-32	Sn	cfg	11	0	3d10	4s1	4p6	4d1	5s1	

50	-32	Sn	cfg	12	0	3d10	4s1	4p6	4d1	5d1
50	-32	Sn	cfg	13	0	3d10	4s2	4p5	4d1	4f1
50	-32	Sn	cfg	14	0	3d10	4s2	4p5	4d1	5p1
50	-32	Sn	cfg	15	0	3d10	4s2	4p5	4d1	5f1
50	-14	Sn	cfg	16	1	4f1				
50	-14	Sn	cfg	17	1	5p1				
50	-14	Sn	cfg	18	1	5f1				
50	-14	Sn	cfg	19	1	6p1				
50	-14	Sn	cfg	20	1	6f1				
50	-14	Sn	cfg	21	1	7p1				
50	-14	Sn	cfg	22	1	7f1				
50	-32	Sn	cfg	23	1	3d10	4s1	4p6	4d1	5p1
50	-32	Sn	cfg	24	1	3d10	4s2	4p5	4d2	
50	-32	Sn	cfg	25	1	3d10	4s2	4p5	4d1	5d1
-1										

Limitations of adas801/adas8#1

- ▶ The Cowan code is well integrated into ADAS workflows.
 - ▶ adas8#1 has been extended to incorporate as optional inputs the U Mons improved structure work.
 - ▶ However.... spin changing transitions are absent.
 - ▶ No resonance effects.
-
- AUTOSTRUCTURE now has a distorted wave module and can generate *adf04* datasets.
 - Auto-generation of input files will be added to the heavy species rules-based routines.
 - Under active development with data already produced (HPS).

adas8#2 — CADW Ionisation

Very similar specification problem as excitation — driven by *adf56* set of rules

<i>index[]</i>	: index of ground configuration of each ion of element in <i>adf56</i> file
<i>config[]</i>	: ground configuration for each ion of element
<i>n_el[]</i>	: number of electrons for each ion of element
<i>no_v_shl[]</i>	: number of shells to treat as valence shells. Max. 2 relevant to relating ion and parent.
<i>v1_shl[]</i>	: first valence shell position in <i>adf56</i> configuration specifications.
<i>v2_shl[]</i>	: second valence shell position in <i>adf56</i> configuration specifications. zero if none defined.
<i>drct_eval_v[]</i>	: evaluate direct ionisation from the valence shell(s).
<i>drct_eval_cl[]</i>	: evaluate direct ionisation from other non-valence (closed) shells.
<i>min_shl_cl[]</i>	: lowest closed shell to include (position in <i>adf56</i> configuration specifications).
<i>exca_eval_v2[]</i>	: evaluate excitation/autoionisation from second valence shell if identified.
<i>max_dn_v2[]</i>	: maximum change in v2 n-shell to be included.
<i>min_dn_v2[]</i>	: minimum change in v2 n-shell to be include.
<i>max_dl_v2[]</i>	: maximum change in v2 l-shell to be included.
<i>min_dl_v2[]</i>	: minimum change in v2 l-shell to be include.
<i>exca_eval_cl[]</i>	: evaluate excitation/autoionisation from other non-valence (closed) shells.
<i>max_dn_cl[]</i>	: maximum change in closed n-shell to be included.
<i>min_dn_cl[]</i>	: minimum change in closed n-shell to be included.
<i>max_dl_cl[]</i>	: maximum change in closed l-shell to be included.
<i>min_dl_cl[]</i>	: minimum change in closed l-shell to be included.
<i>exst_eval[]</i>	: evaluate ionisation from excited states.
<i>exst_adf00_prt[]</i>	: assume parent for building excited states is as present in the <i>adf00</i> data set for the ion.
<i>exst_prt_hole_shl[]</i>	: specify position of shell in ground configuration to form parent if not from <i>adf00</i> above.
<i>max_n_exst[]</i>	: maximum n-shell for excited states to be included.
<i>max_l_exst[]</i>	: maximum l-shell for excited states to be included.
<i>drct_eval_exst_v[]</i>	: evaluate direct ionisation from excited state valence shells.
<i>drct_eval_exst_cl[]</i>	: evaluate direct ionisation from excited state non-valence (closed) shells.
<i>exca_eval_exst_v[]</i>	: evaluate excitation/autoionisation for excited states from valence shells (v1 and v2 above).
<i>exca_eval_exst_cl[]</i>	: evaluate excitation/autoionisation for excited states from non-valence (closed) shells.

adf32 is the driver file for CADW ionisation code from the Auburn group.

At the IDL command line

```
; Add offline-ADAS IDL library to the path  
  
!path = expand_path('/home/adas/offline_adas/adas8#2/idl') + ':' + !path  
  
; Promotion rules - compiled by Adam Foster (arf)  
  
a56file = '/home/adas/adas/adf56/large_arf09.dat'  
  
; Sn13+ !!  
  
adas8xx_ionis_promotion_rules, z_nuc      = 50, $  
                                         z_ion      = 13, $  
                                         a56file   = a56file, $  
                                        adf32     = 'adf32_ca_sn13.dat', $  
                                         comments  = [ 'C-----', '$  
                                         'C I made this!', '$  
                                         'C-----', '$]  
                                         ]
```

```

elem  = Sn
stage = 13
ip_z  = 3193147.3
ip_z1 = 2415629.2
seq   = rb
-----
Type = Direct /number=3/
#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 71. 0.5 0.70
  50 14 sn+13 ground 4d1
  50 15 sn+14 from 4d 3d10 4s2 4p6
  -1
#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 73. 0.5 0.70
  50 14 sn+13 ground 4d1
  50 15 sn+14 from 4s 3d10 4s1 4p6 4d1
  -1
#
200-51 1 2 01. 1. 5.0E-08 1.0E-11-2 0130 0 1.00 0.65 72. 0.5 0.70
  50 14 sn+13 ground 4d1
  50 15 sn+14 from 4p 3d10 4s2 4p5 4d1
  -1
-----
Type = InDirect /number=2/
#
20 -51 0 2 10 1.0 5.e-08 1.e-11-2 130 1.0 0.65 66. 0.5 0.7
  50 14 sn+13 ground 4d1
  50 14 sn+13 via 4d 3d10 4s1 4p6 4d2
  50 14 sn+13 via 4f 3d10 4s1 4p6 4d1 4f1
  50 14 sn+13 via 5s 3d10 4s1 4p6 4d1 5s1
  -
  -
  50 14 sn+13 via 7h 3d10 4s1 4p6 4d1 7h1
  50 14 sn+13 via 7i 3d10 4s1 4p6 4d1 7i1
  -1
#
20 -51 0 2 10 1.0 5.e-08 1.e-11-2 130 1.0 0.65 66. 0.5 0.7
  50 14 sn+13 ground 4d1
  50 14 sn+13 via 4d 3d10 4s2 4p5 4d2
  50 14 sn+13 via 4f 3d10 4s2 4p5 4d1 4f1
  -
  -
  50 14 sn+13 via 7h 3d10 4s2 4p5 4d1 7h1
  50 14 sn+13 via 7i 3d10 4s2 4p5 4d1 7i1
  -1
-----
C-----
C I made this!
C-----

```

At the unix command line

```
/home/adas/offline_adas/adas8#2/adas8#2.pl \
adf32_ca_sn13.dat adf23_ca_sn13.dat
```

Return to IDL to inspect the results

```
read_adf23, file='adf23_ca_sn13.dat', fulldata=all, szd_total=szd
help, szd,/st

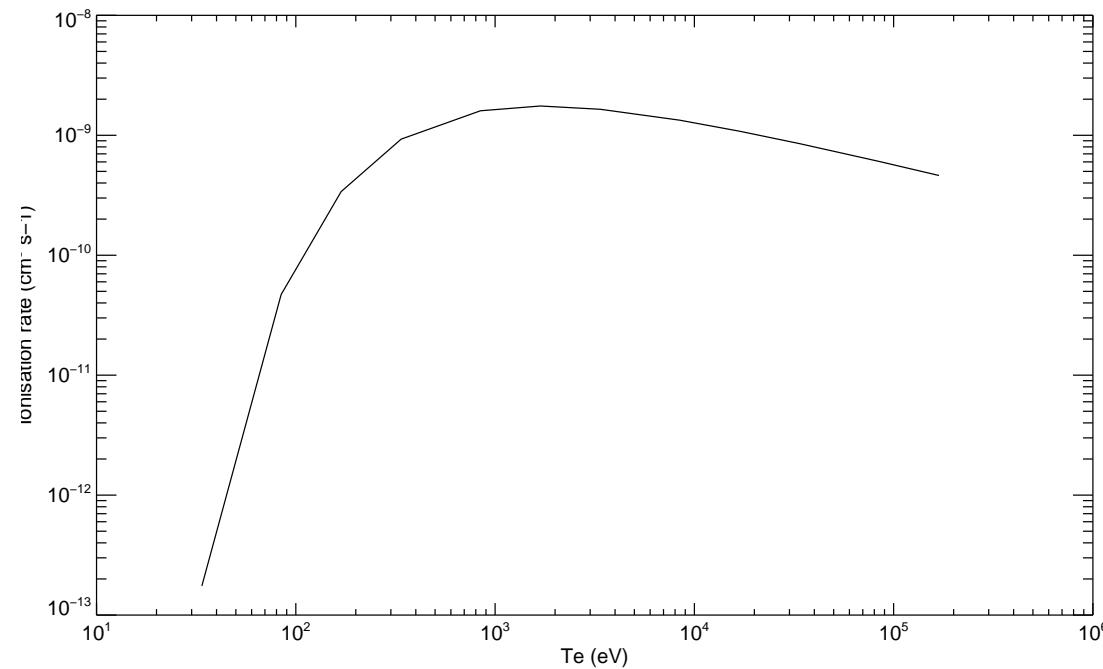
** Structure <a3e784c>, 7 tags, length=6576, data length=6576, refs=1:
    TE           DOUBLE   Array[12]
    Q_ION        DOUBLE   Array[1, 3, 12]
    IS_Q_ION     LONG     Array[1, 3, 12]
    Q_EXC        DOUBLE   Array[1, 41, 12]
    IS_Q_EXC     LONG     Array[1, 41, 12]
    QTOT         DOUBLE   Array[1, 1, 12]
    IS_QTOT      LONG     Array[1, 1, 12]
```

```

te = reform(szd.te) / 11605.0
szd = reform(szd.qtot*10.0^szd.is_qtot) > 1.0e-36

plot_oo, te, szd,           $
    xtitle='Te (eV)', $
    ytitle = 'Ionisation rate (cm!u3!n s-1!n)'

```



Uplift of baseline

As codes mature they can transition from the domain of the specialist to routine, unattended mass data generation. Standard R-matrix is now at this stage — adas8#3.

```
/home/adas/offline_adas/adas8#3/scripts/adas8#3.pl input.dat Z
```

GENERAL	SCALING PARAMETERS
2Jmax_ex = 23	1s = 1.0
2Jmax_nx = 91	2s = 1.0
maxc = 51	2p = 1.0
mesh_fine = 0.0025	3s = 1.0
mesh_coarse = 0.01	3p = 1.0
maxe/ionpot = 3	3d = 1.0
rdamp = 1	
adamp = 0	

CONFIGURATION LIST

```
1s2  
1s1 2s1  
1s1 2p1  
1s1 3s1  
1s1 3p1  
1s1 3d1
```